Clipper

ธาวินทร์ ลีลาคุณารักษ์

คลิปเปอร์
กับการเขียนโปรแกรมตรวจสอบสิทธิผู้ใช้

แนวคิด 3 ข้อ สำหรับการเขียนโปรแกรม เพื่อจะสามารถกำหนดและตรวจสอบ สิทธิผู้ใช้

เวลาที่เราทำงานกับระบบงานใหญ่ ๆ เราจะพบว่าจะต้องมีการบันทึกชื่อผู้ใช้ก่อนเข้าไปใช้งานในระบบ ถ้าเป็นระบบที่ออกแบบมาเป็นอย่างดีก็จะสามารถกำหนดและตรวจสอบได้ว่าผู้ใช้ที่เข้ามาในระบบนั้นมีสิทธิในการใช้งาน ณ จุดไหนของระบบบ้าง

ในการเขียนโปรแกรมเพื่อจะจัดการเกี่ยวกับสิทธิผู้ใช้โปรแกรมนั้น เราจะพบว่ามีเรื่องที่เราต้องพิจารณา 2 เรื่อง คือ

1. เมื่อผู้ใช้ LOGIN เข้าไปในระบบคอมพิวเตอร์แล้ว คอมพิวเตอร์จะตรวจสอบอย่างไรว่า ผู้ใช้คนนี้มีสิทธิ์ใช้งานไหนบ้าง งานไหนที่ไม่อนุญาตให้ผู้ใช้คนนี้เข้าไปทำงาน หรืองานที่ผู้ใช้คนนี้มีสิทธิ์ใช้แต่มีบางจุดที่ไม่อนุญาตให้ทำการแก้ไขคอมพิวเตอร์จะควบคุมอย่างไร

2. การเก็บข่าวสารเกี่ยวกับผู้ใช้ระบบมีวิธีการเก็บอย่างไรบ้าง

ผู้เขียนขอใช้ภาษา XBase โดยใช้คลิปเปอร์เป็นตัวอ้างอิงสำหรับประกอบบทความชุดนี้ ผู้เขียนขอเสนอแนวคิดง่าย ๆ ที่ใช้กัน 3 วิธีคือ

วิธีที่ 1

กำหนดไปเลยว่า ผู้ใช้คนหนึ่ง ๆ ใช้โปรแกรมอะไรได้บ้าง เป็นวิธีที่ง่ายที่สุดในการเขียนโปรแกรมตรวจสอบสิทธิผู้ใช้ โดยทั่วไปเรามักจะตรวจสอบที่เมนูหลักของโปรแกรมอย่างเดียว ตัวอย่างเช่น ในเมนูหลักของเรามีรายการต่าง ๆ ดังนี้

- คลังสินค้า
- จัดซื้อ
- งานขายสินค้า
- ส่วนจัดการระบบ

ถ้านาย ก เป็นพนักงานคลังสินค้า เราจะกำหนดให้เข้าทำงานได้เฉพาะในส่วนของคลังสินค้า

นาย ข เป็นพนักงานจัดซื้อ เราจะกำหนดให้เข้าทำงานได้เฉพาะงานจัดซื้อ

นาย ค เป็นผู้จัดการแผนกซึ่งดูแลคลังและจัดซื้อ ดังนั้นนาย ค มีสิทธิ์เข้าทำงานได้ทั้งในส่วนงานคลังสินค้าและงานจัดซื้อ

นาย ง เป็นผู้จัดการทั่วไป มีสิทธิ์ที่จะดูแลในทุกส่วนของระบบงาน

จากข้อมูลข้างต้นเรานำมาเขียนลงในตาราง ซึ่งจะได้ผลดังตารางที่ 1

จากตารางที่ 1 เราจะนำมาออกแบบแฟ้มข้อมูลที่จะใช้เก็บข่าวสารต่าง ๆ ของผู้ใช้ระบบ ได้ดังตารางที่ 2

ในรูปที่ 1 เป็นจอภาพของโปรแกรมกำหนดสิทธิของผู้ใช้ระบบแต่ละราย ตามแนวความคิดนี้

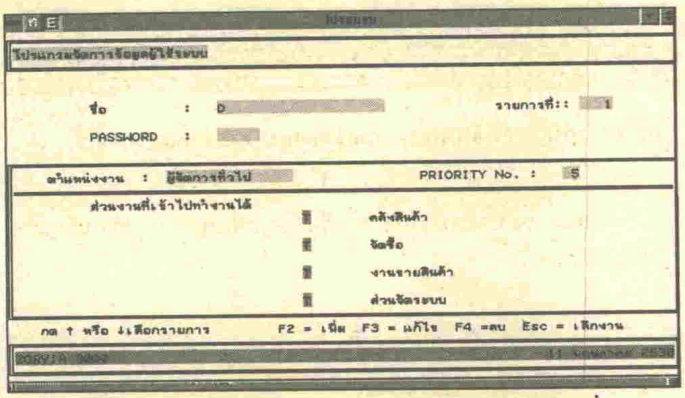
ตารางที่ 1

ระบบงาน				
ผู้ใช้	คลังสินค้า	จัดซื้อ	งานขาย	ส่วนจัดการระบบ
นาย ก	Y	N	N	N
นาย ข	N	Y	N	N
นาย ค	Y	Y	N	N
นาย ง	Y	Y	Y	Y

ตารางที่ 2

Structure for database: C:user1.dbf
Number of data records: 5
Date of last update : 05/10/95

Field	Field Name	Type	Width	Dec
1	NAME	Character	20	เก็บชื่อผู้ใช้
2	PASSWORD	Character	5	เก็บรหัสผ่าน
3	INVEN	Logical	1	เก็บสิทธิการเข้าทำงานส่วนคลังสินค้า
4	PURCHASE	Logical	1	เก็บสิทธิการเข้าทำงานส่วนงานจัดซื้อ
5	SALE	Logical	1	เก็บสิทธิการเข้าทำงานส่วนงานขายสินค้า
6	SYSTEM	Logical	1	เก็บสิทธิการเข้าทำงานส่วนจัดการระบบ
** Total **			30	



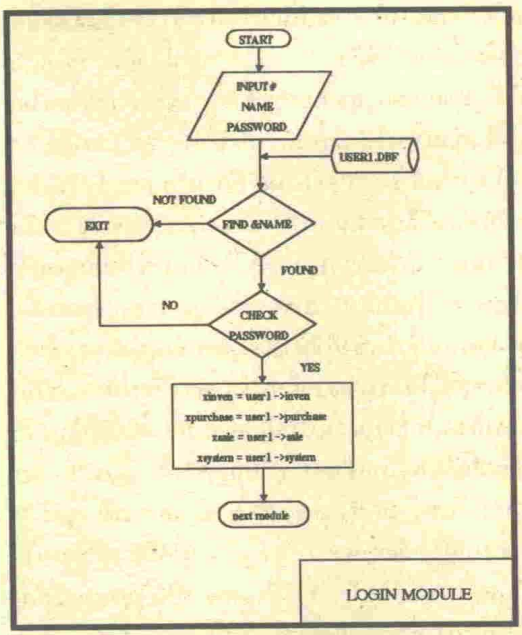
รูปที่ 1 จอภาพของโปรแกรมกำหนดสิทธิของผู้ใช้ระบบตามวิธีที่ 1

ในส่วนการทำงานของโปรแกรม เราจะกำหนดตัวแปรแบบ LOGICAL ตามจำนวนโปรแกรมที่เราต้องการจะควบคุมการเข้าใช้งานไว้ในส่วนของโปรแกรมหลัก อย่างในตัวอย่างที่ 1 นี้ผู้เขียนขอกำหนดตัวแปรดังกล่าวไว้ดังนี้ (ในโปรแกรม TEST1.PRG บรรทัดที่ 8 ถึง 11)

ตัวอย่างที่ 1

```
xinven     = .F.   ใช้ตรวจสอบสิทธิการเข้าทำงานในส่วนคลังสินค้า
xpurchase  = .F.   ใช้ตรวจสอบสิทธิการเข้าทำงานในส่วนงานจัดซื้อ
xsale      = .F.   ใช้ตรวจสอบสิทธิการเข้าทำงานในส่วนงานขายสินค้า
xsystem    = .F.   ใช้ตรวจสอบสิทธิการเข้าทำงานในส่วนจัดการระบบ
```

ค่าของตัวแปรเหล่านี้จะถูกปรับตามสิทธิของผู้ใช้แต่ละคน เมื่อผู้ใช้คนนั้น LOGIN เข้าระบบ ดังแสดงในโฟลว์ชาร์ต LOGIN MODULE



โฟลว์ชาร์ต LOGIN MODULE

รูปโฟลว์ชาร์ตแสดงการเปลี่ยนแปลงของตัวแปรที่เก็บค่าตรวจสอบสิทธิขณะผู้ใช้ login เข้าระบบ

จากโฟลว์ชาร์ตจะเป็นว่า ค่าของตัวแปรข้างต้นจะถูกเปลี่ยนใหม่หลังจากที่ผู้ใช้ login เข้าระบบเรียบร้อยแล้ว เช่น เมื่อนาย ก login เข้าระบบค่าของตัวแปรดังกล่าวจะเปลี่ยนเป็นดังนี้

```
xinven     =   .T.
xpurchase  =   .F.
xsale      =   .F.
xsystem    =   .F.
```

สำหรับจุดที่จะตรวจสอบสิทธิของผู้ใช้ในโปรแกรมระบบงานของเรานั้น ผู้เขียนขอยกตัวอย่างการตรวจสอบง่าย ๆ คือ เราจะเช็คเฉพาะที่เมนูหลักของโปรแกรม เรามาพิจารณาที่ส่วนของโปรแกรมดังตัวอย่างที่ 2 ก่อนนะครับ (อยู่ใน TEST1.PRG บรรทัดที่ 26 ถึง 43)

ตัวอย่างที่ 2

```
.....
.....
026     @  6, 31  SAY     " เมนู หลัก " COLOR "GR+/B+"
027     @  8, 31  PROMPT  " คลังสินค้า "
028     @ 10, 31  PROMPT  " จัดซื้อ "
029     @ 12, 31  PROMPT  " ขายสินค้า "
030     @ 14, 31  PROMPT  " จัดการระบบ "
031     MENU TO x1
032     DO   CASE
033          CASE (x1 = 0)
034     EXIT
035          CASE (x1 = 1).AND.xinven
036          ..........
037          CASE (x1 = 2).AND.xpurchase
038          ..........
039          CASE (x1 = 3).AND.xsale
040          ..........
041          CASE (x1 = 4).AND.xsystem
042          ..........
043          ENDCASE
.....
.....
```

จากส่วนของตัวอย่างที่ 2 จะเห็นว่าในการตรวจสอบสิทธิของผู้ใช้ตามแนวความคิดนี้ เราจะใช้ตัวแปร LOGICAL ที่เรากำหนดให้เป็นค่าตรวจสอบการใช้งานของแต่ละส่วนมากระทำการ .AND. กับเงื่อนไขที่ผู้ใช้เลือก จะเห็นได้ในส่วนของโปรแกรมบรรทัดที่ 035, 037, 039 และ 040

ในลิสติงที่ 1 โปรแกรม TEST1.PRG เป็นโปรแกรมเล็กที่ผู้เขียนจัดเป็นตัวอย่างสนับสนุนวิธีการดังกล่าวข้างต้น บรรทัดที่ 8 ถึง 11 เป็นการกำหนดตัวแปรแบบ LOGICAL ที่ใช้เก็บเงื่อนไขสิทธิของผู้ใช้โปรแกรม บรรทัดที่ 13 โปรแกรม TEST1.PRG จะเรียกโปรแกรมย่อย (FUNCTION) ชื่อ login() ซึ่งในฟังก์ชัน login() จะทำหน้าที่ตรวจสอบรายละเอียดเบื้องต้นของผู้ใช้ เช่น ตรวจสอบชื่อและรหัสผ่าน หากไม่มีปัญหาก็จะเปลี่ยนค่าตัวแปร LOGICAL ที่เก็บค่าสำหรับตรวจสอบสิทธิตามสิทธิของผู้ใช้ที่ถูกเก็บอยู่ในแฟ้มของผู้ใช้บรรทัดที่ 74 ถึง 78 (ในที่นี้คือ USER1.DBF) ในบรรทัดที่ 35,37,39 และ 40 จะเป็นจุดที่ใช้ตรวจสอบว่าส่วนงานที่ผู้ใช้เลือกที่จะเข้าทำงานนั้น ผู้ใช้มีสิทธิ์ใช้งานในส่วนนั้นหรือไม่ และในบรรทัดที่ 113 ถึง 189 เป็นโปรแกรมย่อย (PROCEDURE) ที่ทำหน้าที่เกี่ยวกับการจัดการข้อมูลของผู้ใช้ระบบ

สำหรับวิธีนี้มีข้อดีคือ การเขียนโปรแกรมไม่ยุ่งยากมากนัก แต่ข้อเสียคือ เราไม่สามารถควบคุมผู้ใช้ในรายละเอียดปลีกย่อยบางจุดที่เราต้องการจะควบคุมได้ เช่น ในหน้าจอ (screen) สำหรับปรับปรุงข้อมูลบางจุด เราต้องการให้ผู้ใช้คนหนึ่งทำการปรับปรุงแก้ไขได้ แต่ไม่ต้องการให้อีกคนหนึ่งมาแก้ไข ณ จุดเดียวกันได้ จึงต้องมีวิธีที่ 2

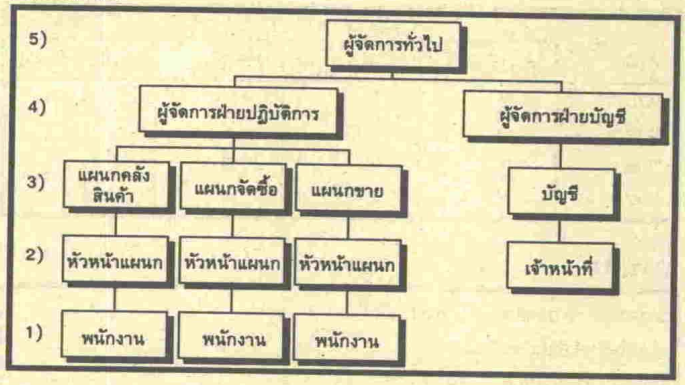
วิธีที่ 2

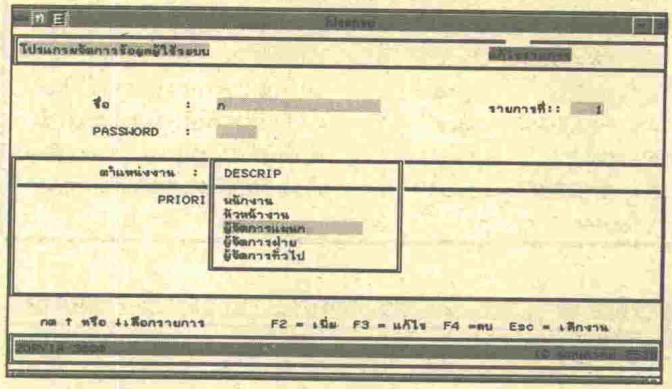
กำหนดสิทธิของผู้ใช้โดยอาศัยโครงสร้างขององค์กร วิธีนี้จะยุ่งยากขึ้นมาอีกขั้นหนึ่งคือ เราต้องกำหนดให้คอมพิวเตอร์รับรู้ว่า ผู้ใช้คนหนึ่ง ๆ อยู่จุดไหนขององค์กร เรามาดู Organize Chart ของกิจการเล็ก ๆ แห่งหนึ่งดังตัวอย่างที่ 3

ในการเขียนโปรแกรมเราคงจะไม่เอารายละเอียดโครงสร้างองค์กรทั้งหมดเข้าไปเก็บในเครื่องคอมพิวเตอร์ แต่เราจะเก็บรายละเอียดบางอย่างที่พอจะสื่อความหมายได้ เช่น เก็บระดับของตำแหน่งงาน ระดับ 1 หมายถึงตำแหน่งพนักงาน ระดับ 2 หมายถึงตำแหน่งหัวหน้างาน เป็นต้น วิธีที่จะช่วยให้เราจัดระดับของตำแหน่งต่าง ๆ ได้ง่ายขึ้นคือ เราใส่ตัวเลขให้ในแต่ละระดับชั้นของโครงสร้างองค์กร ดังใน Organize Chart ตัวอย่างที่ 3

การให้ข่าวสารกับคอมพิวเตอร์ทำได้ง่าย ๆ คือ เราจะบอกคอมพิวเตอร์ว่าผู้ใช้รายนี้มีระดับที่เป็นตัวเลขอยู่ที่ค่าใด (ผู้เขียนขอเรียกตัวเลขชุดนี้ว่า PRIORITY NUMBER(PN)) ในทางปฏิบัติเราจะทำเป็นเมนูสำหรับเลือกระดับตำแหน่งงาน โดยจะให้ผู้ใช้เลือกตำแหน่งงานแล้วนำตำแหน่งงานที่ได้เปรียบเทียบว่าตรงกับเลขใด ดังตัวอย่างในโปรแกรม TEST2.PRG (ใน TEST2.PRG บรรทัดที่ 264 ถึง 354) ในส่วนของการกำหนดรายละเอียดผู้ใช้ระบบ ซึ่งมีหน้าตาดังรูปที่ 2

ตัวอย่างที่ 3





รูปที่ 2 เมนูให้เลือกตำแหน่งงานในการกำหนดรายละเอียดของผู้ใช้

และจากแนวคิดนี้เราจะได้แฟ้มข้อมูลที่เก็บรายละเอียดเกี่ยวกับผู้ใช้ดังตัวอย่างที่ 4

ตัวอย่างที่ 4

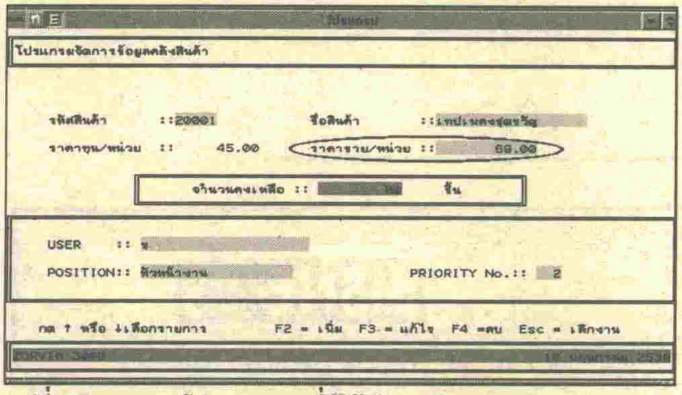
```
Structure for database      : C:user2.dbf
Number of data records      :         5
Date of last update         : 05/10/95
Field  Field Name  Type       Width  Dec
  1    NAME        Character  20     เก็บชื่อผู้ใช้
  2    PASSWORD    Character  5      เก็บรหัสผ่าน
  3    DESCRIP     Character  20     เก็บชื่อตำแหน่งงาน
  4    PRIORITY    Logical    1      เก็บระดับขั้น
                                     ของตำแหน่งงาน
** Total **        48
```

สำหรับข้อมูลในแฟ้มตัวอย่างมีนาย ก เป็นพนักงาน นาย ข เป็นหัวหน้างาน นาย ค ผู้จัดการแผนก นาย ง เป็นผู้จัดการทั่วไป

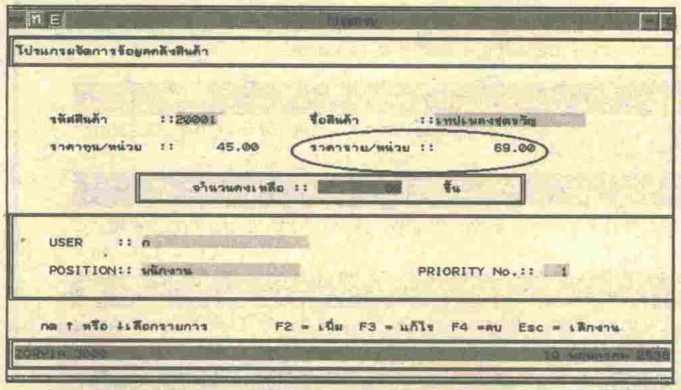
ในการตรวจสอบสิทธิการเข้าใช้งานแต่ละจุดของวิธีที่ 2 เราจะใช้เงื่อนไขตรวจสอบว่าค่า PN ของผู้ใช้ มีค่ามากกว่า (>), น้อยกว่า (<) หรือเท่ากับ (=) ค่า PN ที่เรากำหนดไว้ในโปรแกรม ลองดูส่วนของโปรแกรมดังตัวอย่างที่ 5 (ใน TEST2.PRG บรรทัดที่ 26 ถึง 57)

ตัวอย่างที่ 5

```
.......
.......
026     @  6, 31  SAY     " เมนู หลัก "  COLOR "GR+/B+"
027     @  8, 31  PROMPT " คลังสินค้า "
028     @ 10, 31  PROMPT " จัดซื้อ "
029     @ 12, 31  PROMPT " บัญชี "
030     @ 14, 31  PROMPT " ขายสินค้า "
031     @ 16, 31  PROMPT " จัดการระบบ "
032     MENU TO x1
033     DO   CASE
034          CASE (x1 = 0)
035               EXIT
036          CASE (x1=1).AND.(xpn>=1) &&ระดับหัวหน้างานขึ้นไป
...                .........
038          CASE (x1=2).AND.(xpn> 1) &&ระดับหัวหน้างานขึ้นไป
...                .........
040          CASE (x1=3).AND.(xpn> 1) &&ระดับหัวหน้างานขึ้นไป
...                .........
042          CASE (x1=4).AND.(xpn> 1) &&ระดับหัวหน้างานขึ้นไป
...                .........
044          CASE (x1=5).AND.(xpn>=3) &&ระดับผู้จัดการแผนกขึ้นไป
...                .........
057     ENDCASE
.......
.......
```



รูปที่ 3 จอภาพของโปรแกรมงานที่ผู้ใช้เป็นพนักงาน



รูปที่ 4 จอภาพของโปรแกรมงานที่ผู้ใช้เป็นหัวหน้างาน

ในจุดที่ x1 = 2, x1 = 3 และ x1 = 4 นั้น เราอนุญาตให้ระดับหัวหน้าแผนกขึ้นไปเท่านั้นที่สามารถเข้าไปใช้งานได้ ส่วนที่ x1 = 5 เราให้ผู้จัดการแผนกขึ้นไปเท่านั้นที่มีสิทธิ์ใช้งาน

นอกจากนี้เรายังสามารถควบคุมบางจุดตามที่เราต้องการได้ เช่น ในรูปที่ 3 และ 4 เป็นภาพที่ได้จากโปรแกรม TEST2 (โปรแกรม TEST2.PRG เป็นโปรแกรมประกอบบทความสำหรับวิธีที่ 2 มีแฟ้มข้อมูลที่เพิ่มมาจาก TEST1.PRG คือ organize.dbf ทำหน้าที่เก็บโครงสร้างขององค์กร และ stock.dbf เป็นแฟ้มตัวอย่างเพื่อให้เกิดการทำงานดังภาพที่ 3 และ 4 ตามลำดับ) โดยในรูปที่ 3 เป็นรูปที่ผู้ใช้เป็นพนักงานเข้าใช้งาน และในรูปที่ 4 เป็นโปรแกรมเดียวกัน แต่ผู้ใช้เป็นหัวหน้างานตรงจุดที่วงไว้นั้น ในรูปที่ 3 จะไม่มีแถบแสง ทั้งนี้เนื่องจากเราตั้งเงื่อนไขไว้ให้ระดับหัวหน้าแผนกขึ้นไปเท่าที่ทำการแก้ไขจุดนี้ได้ (ใน TEST2.PRG บรรทัดที่ 141 ถึง 150)

ข้อดีของวิธีนี้คือ เราสามารถควบคุมผู้ใช้ในจุดต่าง ๆ ของระบบงานได้ตามที่เราต้องการ แต่ก็มีข้อเสียคือ วิธีนี้ไม่ได้แบ่งกลุ่มผู้ใช้ที่ชัดเจน ถ้าเราลองใช้โปรแกรม TEST2 จะพบว่านาย ข ซึ่งมีตำแหน่งเป็นหัวหน้างาน แต่เราบอกไม่ได้ว่านาย ข สังกัดหน่วยงานไหน ทำให้นาย ข สามารถเข้าใช้งานได้ทั้งส่วนงานคลังสินค้า จัดซื้อ งานบัญชี และงานขายสินค้า ซึ่งในทางปฏิบัติบุคลากรของหน่วยงานหนึ่งไม่ควรที่จะเข้าไปยุ่งเกี่ยวกับงานของอีกหน่วยงานหนึ่ง

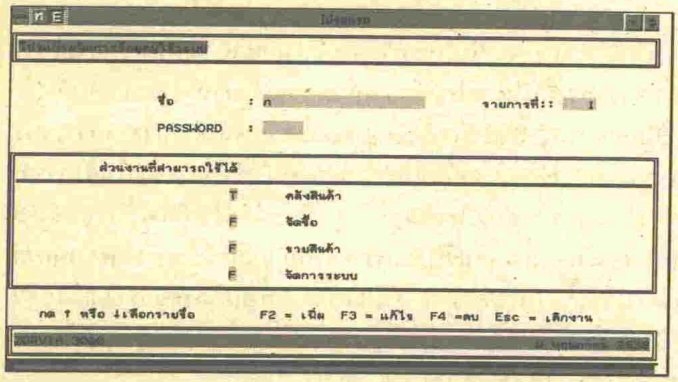
วิธีที่ 3

นำทั้ง 2 วิธีมารวมกัน เป็นการนำข้อดีของทั้ง 2 วิธีข้างต้นมารวมกัน เพื่อลดข้อเสียที่เกิดจากแต่ละวิธี และทำให้แฟ้มข้อมูลที่เก็บรายละเอียดของผู้ใช้เปลี่ยนไปเป็นดังตัวอย่างที่ 6

ตัวอย่างที่ 6

```
Structure for database: C:user3.dbf
Number of data records:     5
Date of last update    : 05/11/95
Field  Field Name  Type       Width  Dec
    1  NAME        Character  20     เก็บชื่อผู้ใช้
    2  DESCRIP     Character  20     เก็บตำแหน่งงาน
    3  PASSWORD    Character  5      เก็บรหัสผ่าน
    4  PRIORITY    Logical    1      เก็บระดับขั้นของตำแหน่งงาน
    5  INVEN       Logical    1      เก็บค่าสิทธิการเข้าใช้งานส่วนคลังสินค้า
    6  PURCHASE    Logical    1      เก็บค่าสิทธิการเข้าใช้งานส่วนจัดซื้อ
    7  SALE        Logical    1      เก็บค่าสิทธิการเข้าใช้งานส่วนขายสินค้า
    8  SYSTEM      Logical    1      เก็บค่าสิทธิการเข้าใช้งานส่วนจัดการระบบ
** Total **                   52
```

ส่วนหน้าตาของโปรแกรมจัดการข้อมูลผู้ใช้ระบบก็จะมีหน้าตาดังรูปที่ 5



รูปที่ 5 หน้าตาของโปรแกรมจัดการข้อมูลผู้ใช้ตามวิธีที่ 3

โปรแกรม TEST3.PRG เป็นตัวอย่างโปรแกรมที่แสดงการทำงานของวิธีที่ 3

ในการตรวจสอบสิทธิของผู้ใช้ ณ จุดต่าง ๆ ในโปรแกรม เราจะใช้หลักการของวิธีแรกในการควบคุมการเข้าทำงานของแต่ละหน่วยงาน (บรรทัดที่ 26 ถึง 54 ของ TEST3.PRG) และเราใช้หลักการของวิธีที่ 2 ในการควบคุมการแก้ไขข้อมูลบางจุด (บรรทัดที่ 116 ถึง 202 ของ TEST3.PRG)

สำหรับการทำงานก็จะคล้ายกับโปรแกรม TEST1 ที่เมนูหลัก และคล้ายโปรแกรม TEST2 การควบคุมการแก้ไขข้อมูลในบางจุด

ในชีวิตการทำงานจริงนั้น ผู้จัดการขึ้นไปคงไม่มีเวลาที่จะมาบันทึก ปรับปรุง หรือแก้ไขข้อมูลด้วยตัวเองต้องมอบหมายให้คนอื่น ๆ ทำงานแทน ในส่วนนี้ผู้เขียนขอฝากไปคิดเป็นการบ้านนะครับ สำหรับท่านที่ต้องการซอฟต์เมล์ไว้ศึกษาขอให้ติดต่อทางกองบรรณาธิการวารสารไมโครคอมพิวเตอร์ โอกาสหน้าพบกันใหม่ครับ

GROW WITH THE MARKET LADER IN
INFORMATION
TECHNOLOGY

BISNEWS
THE POWER OF INFORMATION

Established as the lading provider of PC-accessed business information servicces in Thailand, we provide real-time news and market data from around the world and Thailand to thousands of customers. To keep pace with our growth we require forward-thinking confident individuals who are determined to learn about and stay on top of technology as it advances, We provide a flexible working environment with attractive remuneration and excellent fringe benefits. Positions required:

System Egineer
Bachelors degree or higher in Electronic or Teleommunication Engineering, Knowledge and experience in networking, broadcasting, datacommunication and PC skills preferable. Profic.ent written and spoken English.

Software Engineers/Programmers
Bachelors degree or higher in Computer Science or related field. Familiar with Windows Environment, experience in any programming language.

Software Design Engineers
Bachelors degree or higher in Computer Science or related field. Familiar with widows programming, experience in C/C++ programming languages, event-driven, object oriented programming and TCP/IP or Winsock programming preferable.

Technical Support
Diploma or higher in Electronics or related field. PC skills, experience is an advantage.

Applications stating qualifications and experience as well as current and expected remuneration together with a recent photograph should be sent to:

Personnel Manager
Bisnews Public Company Limited
3rd Floor Mahatun Plaza Building 888/178 Ploenchit Road, Bangkok 10330

FIVE-DAY WORK WEEK